I thank the reviewer for their careful reading of my manuscript and for their constructive comments and suggestions. In the following, I respond to each comment point by point. Reviewer comments are reproduced in full (RC), followed by my author responses (AC). All revisions indicated will be incorporated in the revised manuscript.

RC1: This manuscript by Confidence Duku presents results of a model comparison on flood predictions in various rivers around the world. I had a very hard time understanding parts of the paper that made reviewing the study in every detail impossible for me. Therefore, I will only concentrate on some main points before eventually looking more into details in a revised version.
AC1: The reviewer states that they had a hard time understanding parts of the paper but does not specify which sections. This is disappointing, as the purpose of peer review is to provide constructive feedback that can help the author identify unclear aspects and improve the manuscript. Without concrete indications, it is impossible to respond to or correct the issues the reviewer alludes to. That said, I will still revise the manuscript with the aim of improving overall clarity, particularly in the Methods and Discussion, but I would have expected more specific guidance to address this comment effectively.

RC2: Bakaano-Hydro modelling framework What is the Bakaano-Hydro modelling framework? Almost no details are given in this manuscript and not even a reference to the seemingly connected preprint https://egusphere.copernicus.org/preprints/2025/egusphere-2025-1633/ is provided. Since the Bakaano-Hydro modelling framework is not an established thing, I expect this paper to at least contain a reference to the modeling paper but still list some details here. E.g. what are the input data (and e.g. for the weather data, from which product/provider)?
AC2: The reviewer states that "almost no details are given" about the Bakaano-Hydro modelling framework. This is not correct. The methodology section (pp. 5–8) explicitly describes the three-step structure of the framework (runoff generation, routing, and temporal sequence modelling) and provides details on each step, including the meteorological and hydrological inputs. If the reviewer finds the description or structure difficult to follow, that is understandable and I will revise for clarity. However, to say that "almost no detail is given" overlooks content that is already in the manuscript.
Regarding the missing reference to the GMD preprint, both manuscripts were submitted around the same time and at submission the preprint was not yet available. The HESS paper was written as a stand-alone study, with sufficient methodological description and data information to ensure reproducibility. That said, in the revised version I will cite the

GMD preprint (Duku, 2025: https://egusphere.copernicus.org/preprints/2025/egusphere-2025-1633/) and add a clearer upfront description of the framework and input datasets to strengthen coherence and provide additional context for readers unfamiliar with Bakaano-Hydro.

RC3: From what I understand, the framework consists of three steps: 1) VegET, 2) routing, and 3) the TCN based neural network. Since I assume also part 1 and part 2 contain some model parameters, how are they calibrated? Since e.g. part 1 is still without routing, against which observations is this part calibrated? What periods are used for calibrating part 1 and part 2? Since part two is already a routed discharge, how well does this routed discharge perform against the observations from GRDC? How much additional benefit is gained by post-processing the routed discharge with the TCN?

AC3: The reviewer assumes that the first and second phases of my framework (runoff generation using VegET and subsequent routing) involve calibration, which is not the case. These two steps are not calibrated against discharge observations. Instead, they serve as intermediate stages that capture the spatiotemporal dynamics of runoff and routed flow. The outputs are not intended to represent observed discharge directly, but to provide physically meaningful inputs into the TCN component.

In particular, the routed flow is not treated as discharge in my framework. All routed runoff is made available at the outlet on the same day, whereas in reality transmission losses, flow delays, and storage effects occur. Thus, the routed flow serves as a structured biophysical signal for the neural network, not as a final calibrated product. The calibration in my framework occurs only in the TCN component, which learns to map these intermediate signals to observed discharge.

I will revise the methodology section to make this distinction clearer and will explicitly note that VegET runoff generation and topographic flow routing are uncalibrated intermediate steps. I will also expand the explanation of why routed flow is not evaluated directly against GRDC discharge, and how the additional benefit of the TCN is precisely to transform these intermediate signals into realistic discharge predictions.

RC4: In different places of the manuscript it sounds like the neural network part of the model is responsible for simulating streamflow. E.g. P7 L10: “The final phase of Bakaano-Hydro involves the application of a deep learning model to simulate streamflow.” Isn’t part two already outputting routed streamflow? From my limited understanding, the TCN layer gets as input routed streamflow (converted to mm/d by dividing through the upstream area) and depths-to-water-index. How can this part be responsible for simulating streamflow?

Isn't the neural network part simply a post-processor for the VegET+routing model?
AC4: I thank the reviewer for raising this point. To clarify: in my framework, the TCN is indeed responsible for simulating the final streamflow that is compared with GRDC observations. VegET produces runoff, which is not discharge, and the subsequent routing step produces routed runoff, which I treat as an intermediate product rather than streamflow. This is because my routing implementation does not account for transmission losses, storage, flow delays, or other attenuation processes. Instead, it makes all routed water available at the outlet on the same day, which overestimates flow and distorts timing relative to observed discharge.
For this reason, the routed runoff is not considered a final discharge product but an input feature to the TCN. The TCN is trained to transform this intermediate, hydrologically structured signal—together with other predictors such as depth-to-water index—into realistic discharge simulations. I will revise the Methods section to clarify this hierarchy.

RC5: What are the 3 dynamic inputs that are passed to the neural network, as mentioned on P7 L 27. The following sentences only list the routed streamflow and the depth-to-water-index.
AC5: I appreciate the request for clarification. Routed flow is the core dynamic signal passed into the neural network, but for training stability and to capture different hydrological perspectives I transform it into three related dynamic inputs. These are:

1. Raw routed flow (untransformed), representing the aggregate upstream contribution.
2. Routed flow normalized by catchment area (i.e., divided by the number of upstream grid cells), which approximates a depth-equivalent representation and reduces scale effects.
3. Depth-to-water index (DTW), which captures spatial variability in groundwater accessibility and subsurface flow potential.
   Together, these provide the three dynamic inputs. I will revise the text to spell out these inputs explicitly so that readers do not confuse routed flow with only a single feature.

RC6: Why has the neural network part two different branches, once where streamflow is normalized by the upstream area and once where it is normalized by max simulation?
AC6: The reviewer's description of the neural network branches is inaccurate. As stated in the manuscript (p. 7, lines 27–28), the two branches process different types of inputs: one branch processes dynamic inputs (routed runoff and related signals), while the other

processes static catchment descriptors. These branches are then fused within the network to jointly inform streamflow simulation. The motivation for this architecture is to explicitly separate temporally varying features from fixed physiographic characteristics, which is a standard design choice in hydrology-informed neural networks. This separation improves interpretability and ensures that static catchment information is not conflated with dynamic meteorological–hydrological drivers.
I do not consider this design to be a result of trial-and-error; it follows from the conceptual need to treat static and dynamic features differently. While I agree that ablation studies can provide insights into the marginal contribution of architectural components, they are not the focus of this paper, which is positioned as a benchmarking study. My objective is to evaluate whether a hydrology-guided hybrid framework improves flood forecasting reliability compared to established models, rather than to isolate the effect of each architectural choice. I will, however, revise the Methods section to make the role of the two branches clearer.

RC7: I strongly recommend splitting Section 2.1 into a dedicated data section and a dedicated model section. For the data part further, I recommend adding a table to the manuscript that clearly shows which input features are being used with columns for: name, units, source, time periods. Another point is that by choosing yet another weather dataset, it is hard to know how much of the performance differences can be attributed to the modeling framework or the quality of the model input data. Since the presented setup is not suitable for an operational setting (see above) it begs the question what the main focus of this study is? When focussing on the comparison of models, it is usually advised to use the same input data sources for all models to eliminate the impact of data quality on the results. If the focus is comparing flood forecast systems (as in Nearing et al., 2024), then obviously each model can use anything, with the caveat that it should be available in a real operational context.
AC7: I agree with the reviewer's recommendation. In the revised manuscript, I will restructure Section 2.1 into two dedicated subsections: one focusing on the data and one on the model framework. In the data subsection, I will add a summary table listing all input features.

RC8: For the meteorological data, what was the reason for choosing to use a (bias adjusted) climate model rather than a meteorological dataset (either reanalysis, hindcast, or historical forecast model output)? Since this paper compares two different operational forecasting models and in various places suggests that the presented framework improves

upon these operational systems, it is probably worth noting that the model, with the data from this study, could not run operationally.

AC8: My intention in using bias-adjusted climate model data was to demonstrate the capability of Bakaano-Hydro to operate with high-resolution climate datasets. That said, I welcome the reviewer's suggestion and will revise my experimental setup to include reanalysis data as forcing, which will also demonstrate that the framework can run with the kinds of meteorological datasets typically used in operational forecasting.

RC9: Operational flood forecasting models don't alert if simulations surpass thresholds computed from observations but rather if simulations surpass thresholds based on simulations, as it also has been done in Nearing et al. (2024) for GloFAS and the Google Flood Forecasting model. This removes the bias problems of simulations vs. observations and could lead to perfect flood alert rates, even if the model has a constant bias problem. Why is this even more important here? In this study, the author presents a model that performs essentially per-gauge bias correction (by inputting the routed streamflow into the TCN layer and fitting the TCN layer against observations) and then compares this to two globally calibrated operational models.

AC9: I acknowledge the reviewer's point that in operational forecasting systems, thresholds are typically computed from simulations rather than observations. This is done to avoid systematic biases in model magnitude and to provide internal consistency within each system. However, the purpose of this paper is not to demonstrate an operational forecasting configuration, but to provide a fair benchmarking comparison across different models. Using simulated outputs to compute thresholds independently for each model would not provide a fair basis for comparison, because models with systematic low or high bias in peak streamflow would effectively be evaluated on timing alone. For example, a model that consistently underestimates peak flows could still achieve "perfect" alert rates if its own underestimated simulations were used to define thresholds. This would artificially inflate its performance relative to a model that better matches observed flood magnitudes. By defining thresholds based on observed discharge, I ensure that both timing and magnitude are evaluated consistently across models. This approach avoids bias corrections being "baked into" the threshold definition and enables an objective assessment of which model provides more reliable flood forecasts in relation to reality. This benchmarking choice follows the reasoning that improvements in skill should be attributed to model behavior, not to calibration of evaluation metrics.

Having said that, I will take the reviewers comment into consideration in the revised version

RC10: Second: P12 L 12 “CSI, POD and FAR values for flood events with 1-, 2-, 5-, and 10-year return periods and 0-, 1- and 2-day timing tolerance were calculated using simulated data from Bakaano-Hydro for the period 1982 to 2016” This sentence suggests that the majority of the evaluation period for which you computed metrics are the training period of the Baakano-Hydro model (P8 L9 “The training period covered 1989–2016”). For the Google Flood Forecasting model, even for the full-run, all simulations are the result of a temporal k-fold cross validation, as stated in Nearing et al. (2024), i.e. all simulations are “unseen test data”. If my understanding is correct, this renders large parts of the evaluation/conclusion unfair, as per-gauge bias corrected training data is compared to test period data from a globally calibrated model with the additional caveat from the point above.

AC10: I thank the reviewer for raising this point. It is correct that the metrics reported in the manuscript were computed over the full 1982–2016 period, which includes both the training (1989–2016) and validation (1982–1988) subsets for Bakaano-Hydro. This approach follows the evaluation setup used in GloFAS (Alfieri et al., 2013; Harrigan et al., 2023), where performance is reported over the full hindcast period, and was adopted here to allow direct comparability with that benchmark.

I acknowledge, however, that this creates an asymmetry with the Google Flood Forecasting model, for which all simulations are generated out-of-sample via temporal cross-validation (Nearing et al., 2024). This means the evaluation is fair with respect to GloFAS but less fair with respect to the Google AI model.

To address this, I will revise my experimental setup by introducing a cross-validation procedure for Bakaano-Hydro, ensuring that all streamflow simulations used for computing CSI, POD, and FAR are strictly out-of-sample. This will place Bakaano-Hydro on equal footing with the Google AI model while also maintaining comparability with GloFAS.

RC11: The metrics are not well explained or referenced. Since I wouldn’t consider CSI, POD, FAR super common metrics in modeling papers of hydrology, I would recommend adding an explanation of these metrics, probably the equations, and also mention the range of these metrics and their optimal values.

AC11: I respectfully disagree with the reviewer’s assessment that CSI, POD, and FAR are not “super common” in hydrological modeling papers. These metrics are in fact widely used in event-based flood and precipitation verification studies (e.g., Gründemann et al., 2018; Yang et al., 2021), including the evaluation of GloFAS and other global flood forecasting systems. They are standard contingency-table metrics in meteorology and hydrology for quantifying detection skill, false alarms, and overall critical success. Because these metrics are well established in the flood forecasting literature, I did not

originally include their equations. However, to avoid any ambiguity for readers who may be less familiar with event-based verification, I will add brief definitions and references in the Methods section. I do not consider it necessary to provide full derivations, since these are readily available in the cited literature, but I will ensure that interpretation and optimal ranges (e.g., POD and CSI closer to 1, FAR closer to 0) are clearly stated.

RC12: Figure 2: The diverging color scale, centered around 0 and with range (-1,1) does not make sense for all metrics, E.g. Alpha-NSE and Beta-KGE have their optimal value at 1. Also, please indicate what are the optimal values for each metric, which facilitates interpreting the results.
AC12: I agree with the reviewer and will revise the figure and the captions accordingly.

RC13: Figure 3/4/5: If the whiskers show the full range, why are there points outside of the whiskers? Are the whiskers maybe just indicating a certain percentile?
AC13: In my RainCloud plots, the whiskers do not represent the full data range. They extend only to the most extreme values within 1.5 × IQR of the lower and upper quartiles, following the standard boxplot definition. Points lying outside of the whiskers are outliers. Since all individual basin-level values are plotted in the raincloud strip, these points are explicitly shown rather than hidden, which explains why many fall outside the whiskers. I will clarify this in the figure captions.

RC14: Figure 6/7: They feel a bit too much and/or not very well presented, as indicating the median of model performance over all stations in a river basin only by color makes it hard for me to really get any details out of these figures. I understand the reason behind these figures (showing model differences across spatial patterns) but maybe you find a better way to visualize the results. Maybe on a map? But two pages of colored rectangles feels like an overkill in the main paper.
AC14: I agree with the reviewer and will revise the figure and the captions accordingly.

RC15: On a more general note: Figure 3/4/5/6/7 all have three columns for different flood timing tolerance. I wonder if all three of these columns for each figure are needed in the main paper or if some of the plots could be moved to the appendix. In my opinion, there are not dramatically different patterns between the columns that make it necessary to have all

three columns in the main paper.
AC15: I agree with the reviewer and will revise the figure and the captions accordingly.

RC16: P12 L9 "…using the annual maximum series (AMS) method" This sounds like it is missing a reference.
AC16: I will add a reference for the Annual Maximum Series (AMS) method to ensure proper attribution.

RC17: P12 L10f As far as I know, the paper by Nearing et al. compares GloFAS and the Google Flood forecasting model on their true forecast skill under operational settings (i.e. using only data that is available in real time) and evaluating the skill at different lead times.
AC17: The reviewer is correct that Nearing et al. (2024) evaluated GloFAS and the Google Flood Forecasting model under operational conditions across multiple lead times. My study is most directly comparable to their 0-day lead time (nowcast) evaluation, which is essentially a hindcast assessment based on data available up to the same day. At this lead time, both approaches assess retrospective model skill in reproducing observed flood events. I will revise the text to make clear that my evaluation relates specifically to the 0-day results in Nearing et al., and does not extend to their multi-day forecast skill analysis.

RC18: P22 L4 "Importantly, Bakaano-Hydro advances the state of data-driven hydrological modeling by embedding physically meaningful processes within a neural network architecture." I think I disagree with this conclusion. How was the state of data-driven hydrological modelling advanced by the findings of this paper? The presented framework does not embed physically meaningful processes in a neural network. The neural network gets the output of a physically inspired model as input. This is something fundamentally different than "embedding physically meaningful processes within a neural network."
AC18: I thank the reviewer for this critical point. I agree that my wording overstated the claim. The Bakaano-Hydro framework does not embed physically meaningful processes within the neural network itself. Rather, it integrates outputs from a process-based runoff model and routing scheme as structured inputs into the neural network. I will revise this sentence to clarify that the contribution is in combining process-based hydrological modeling with neural network sequence modeling in a hybrid framework, demonstrating improved reliability in data-scarce regions.